

WT_PERF USER'S GUIDE

by

Marshall L. Buhl, Jr.
National Wind Technology Center
National Renewable Energy Laboratory
Golden, Colorado

November 21, 2000

INTRODUCTION

I wrote WT_Perf because I needed several things from PROP-PC that it just didn't supply. One was to be able to parametrically vary RPM. The other was output in a format that was easy to import into Excel. I also hated the input file format because I always had to refer to the manual just so I could read the file.

I started adding some of these features to PROP-PC and found that I was also having difficulty reading the code. I started renaming variables and cleaning up the programming (nasty stuff, it was). I found myself optimizing some of the calculations and decided that I may as well completely rewrite the program.

I found a few smallish bugs in PROP-PC, fixed them, and then added some enhancements (see "change.log"). I also rewrote many of the algorithms. I really don't grok the non-dimensional stuff, so instead of doing the computations in non-dimensional form and converting to dimensional data, I did it the other way around.

As you can see, WT_Perf is a completely different program than PROP-PC, so I feel comfortable in renaming it. Also, there is no longer any propeller code in the program, so "PROP" really seems very inaccurate. They *do* give similar results--especially if the tip-loss and 3D options are disabled.

I hope I've made WT_Perf a very user-friendly program. I put a lot of effort into that feature and also tried to make it very portable. I compiled WT_Perf with Compaq Visual Fortran v6.1A. It should work on any 32-bit Windows PC.

RETRIEVING FILES FROM THE ARCHIVE

You should download the WT_Perf archive from our web server page <http://wind2.nrel.gov/designcodes/wtperf>. The file should have a name something like "WT_Perf_v220.exe." Create a WT_Perf folder somewhere on your file system and put this file there. You can double click on it from Windows Explorer or by entering "WT_Perf_v220" at

a command prompt with the WT_Perf folder as the current directory. This will create some files and folders.

DISTRIBUTED FILES

The files included in the archive of WT_Perf are as follows:

ArcFiles.txt	The list of files that are written to the archive.
Archive.bat	The batch file that creates the archive.
Change.log	The list of changes to WT_Perf.
PROP2WTP.exe	The program used to convert old-style PROP-PC files to WT_Perf format.
WT_Perf.doc	This user's guide in Word format.
WT_Perf.exe	The WT_Perf executable file.
WT_Perf.pdf	This user's guide in PDF format.
DATA\Test.in	A sample input file.
DATA\Test.out	A sample output file.
Source*.*	The source-code files for WT_Perf and PROP2WTP.

USING WT_PERF

To syntax for WT_Perf is:

WT_Perf <input file> <output file>

If you do not enter both arguments, WT_Perf will tell you so and display the syntax for you.

CREATING THE INPUT FILE

Instead of creating an input file from scratch, please copy and edit the example file "Test.in" that is located in the Data folder. Please do not add or remove any lines, except for the variable-length tables such as the blade layout or the airfoil coefficients. Please do not

depend on the values found in this file to be accurate representations of the Unsteady Aerodynamics Experiment turbine. A section-by-section description of the input file follows. In it, variable names use the Letter Gothic typeface. All "flag" variables have values of either True or False.

Header

The first line of the file says simply what type of file it is, you may change the line, but do not remove it or add additional lines.

Job Title

You have one line to describe the turbine being modeled. WT_Perf writes this line in the header of the output file.

Input Configuration

The ECHO_INP flag tells WT_Perf whether or not to echo the input data to the file "echo.out." If you set it true, WT_Perf will write out the input values next to descriptions of them. If WT_Perf crashes due to an input error, checking this file will tell you where things went wrong.

If set to true, the DIMEN flag tells WT_Perf to expect dimensional input parameters. If false, then parameters like the chord are assumed to be non-dimensional. If you want to use non-dimensional input, divide parameters such as the chord by the rotor radius. The comments in the input tell you which parameters can be non-dimensional. If a parameter can be normalized by the rotor radius, the comment will mention so in the units for the parameter. For instance, the units for the hub radius are "[length or div by radius]," where the word "length" means the parameter has units of length if dimensional, or it is normalized by dividing its length by the radius of the rotor.

The METRIC flag tells whether or not English units are used or if Metric units are used. This parameter does not apply to the wind speed units. The setting for wind speed is not affected by the METRIC flag. Units used are as follows:

Measurement	Metric	English
time	seconds	seconds
length	meters	feet
mass	kilogram	slugs
force	newtons	pounds
angle	degrees	degrees

Output Configuration

Setting the TABDELIM flag to true will tell WT_Perf to generate output files with tabs between the columns instead of used fixed format. Tab-delimited files are

best for importing into spreadsheets, while fixed-format files are best for viewing with a text editor or for printing.

Use the WRITE_BE flag to tell WT_Perf to generate files of blade-element data. For each case run, WT_Perf will generate a line of data for each blade station and sector. If you use, say, four sectors, WT_Perf will generate four lines of data for each station. Each test case will have a separate block of data with a header that describes the columns.

WT_Perf varies up to three parameters in each run. They are the rotor speed in rpm, the blade pitch in degrees, and the wind speed. The wind speed can be entered as a tip-speed ratio or an actual wind speed. The parameters PAR_ROW, PAR_COL, and PAR_SHT. The determine how the output data are tabulated. If all three parameters are varied, then WT_Perf will generate multiple tables of data. What is varied from row to row in the tables is determined by the PAR_ROW parameter. The PAR_COL parameter determines what varies from column to column in the tables. The PAR_SHT parameter determines which of the parametric values varies from sheet to sheet (table to table).

The last five parameters in this section tell WT_Perf which of the possible output values should be written to the output file. They are rotor power (kW), Cp, rotor torque (N-m or ft-lbf), flap bending moment at the hub center (N-m or ft-lbf), and rotor thrust (N or lbf).

Model Configuration

N_SEG tells WT_Perf how many data points along the blade (evenly divided from the center of rotation) will have input parameters specified for them. The input data should be for the centers of the segments. That is, for N_SEG=10, input parameters should be defined for 5%, 15%, ..., 95% of the blade radius (preconed along the blade).

SEG_FRST and SEG_LAST tell the code which of those blade stations should be used in the analysis. For instance, if the 5% point lies within the hub radius, you may want to exclude that point from the calculations. If so, set SEG_FRST to 2.

N_SECT is the number of pie-wedge sectors around to rotor disk that will be used in the calculations. If you set both the TILT (shaft tilt *or* yaw) and SHR_EXP (wind-shear exponent) to zero, there is no need for more than one sector. That is because the all the calculations in all sectors would be the same anyway. WT_Perf actually ignores this parameter in this situation and analyzes one sector. If either of those two parameters is not zero, WT_Perf will use a minimum of four sectors in the analysis.

WT_Perf uses MAX_ITER to set a maximum on the number of iterations to use to converge in the induction loop. The way the code works is that it assumes there is no induction effect. It then calculates the angle of attack and looks up the lift and drag for that angle. It then uses the lift and drag to compute the induced velocity. Using the new values for the induction, it repeats the process. When the induced velocity changes less than a given amount from one iteration to the next, it exits the induction loop. If, after MAX_ITER iterations, the induced velocity changes by more than the given amount, the code gives up and sets all output values to a negative number with all nines in it (for example, the power is given as -999.99 kW).

Algorithm Configuration

If your aerodynamic input data does not cover all angles of attack that will be tried in the induction loop, you should turn on the VITERNA flag so that WT_Perf can use the Viterna method to predict post-stall aerodynamics using flat-plate theory. You really should use measured lift and drag data in the post-stall region, but this can help you in a pinch.

The TIP_LOSS and HUB_LOSS flags tell the code to turn on the Prandtl tip and hub loss algorithms. This should usually be enabled.

The SWIRL flag tell the code to enable the algorithms for the calculation of the tangential induction factor (swirl). This should generally be set to true.

WT_Perf uses the ADV_BRK flag to use the advanced brake state algorithm instead of the classic momentum brake-state model. I usually use the advanced brake-state model.

Do not enable the ADD_3D flag. I added to WT_Perf the 3D aerodynamics calculations from "The Influence of 3D Effects in Lift and Drag on the performance of a stalled Horizontal Axis Wind Turbine Rotor", by Bjorn Montgomerie of ECN, NL. *Bjorn has since told us that the algorithms do not work correctly and should not be used.*

Parametric Analysis Configuration

This section tells WT_Perf what sort of parametric analysis to run. The PIT_ST, PIT_END, and PIT_DEL values define the start, end, and delta pitch angles to use. They are input in degrees. See the discussion of the TIPTWIST below to understand the definition of blade pitch.

The OMG_ST, OMG_END, and OMG_DEL parameters define the start, end, and delta rotor speed in rpm.

When specifying the parametric wind speeds, you can either input values in tip-speed ratio (speed of the blade tip divided by the wind speed), or actual wind

speeds. If you enable the INP_TSR flag, WT_Perf will expect the following line to be tip-speed ratios. SPD_ST, SPD_END, and SPD_DEL define the start, end and delta speed. If INP_TSR is false, enter actual wind speeds. The UNITS string defines the units for actual wind speeds. Three possible values are valid: "mps" will tell the code that the wind-speed values are in meters/second, "fps" will indicate that they are in feet/second, and "mph" will indicate that they are in miles/hour. If INP_TSR is true, this parameter is ignored.

Turbine Data

NUM_BL is the number of blade on the turbine. It must be an integer greater than zero.

ROT_RAD is the rotor radius. It is the distance along the precone blade, so it is a number larger than the swept radius if the precone is not zero.

The TIPTWIST is a tricky number. I added this parameter because I like to think of the pitch angle as being the angle between the tip chord line and the plane of rotation. Others like to think of it as the angle between something like the 70%-radius chord line and the plane of rotation. For either, set the TIPTWIST to the value of the twist at the tip (not necessarily the last data point) of the blade, and pitch should work the way you want it to. The TIPTWIST defines what the pitch angle is—it does not depend on the pitch angle.

The next part of this section contains N_SEG lines defining the chord, thickness, and twist distributions. The first line is for the first blade station (5% for N_SEG=10). If you are entering data in non-dimensional form, divide the chord by the rotor radius (ROT_RAD) and divide the thickness by the chord. Enter the twist in degrees.

The next line is for the hub radius. Enter it in either meters or feet if using dimensional data. Otherwise, divide the hub radius by the rotor radius.

The precone in degrees comes next. Make it positive for downwind precone regardless of whether or not the turbine is downwind or upwind.

The next entry is the shaft tilt in degrees. If you do not use wind shear (SHR_EXP=0), you can also use the shaft tilt for the analysis of off-yaw conditions.

The hub height follows. If using non-dimensional input, divide it by the rotor radius.

Aerodynamic Data

The air density is always entered as a dimensional number. Use either kg/m³ or slugs/ft³. For Standard Temperature and Pressure at sea level, use either 1.225 kg/m³ or 0.00238 slugs/ft³.

The next line is the exponent of the power-law wind shear. For the standard $1/7^{\text{th}}$ power law, use 0.143.

The next entry (SEP_TABL) is a flag that tells WT_Perf whether or not you will enter the lift and drag tables as two separate two-column tables (alpha, lift and alpha, drag) or as a single three-column table (alpha, lift, drag).

The next line has three entries. The first is the segment number, SEG. Segment numbers must be in order and numbered from one to N_SEG. The second entry (NCL) is the number of data points in the lift table. If you want to reuse data from a previous segment, set NCL to zero. In that case, the following line should be another line of SEG, NCL, and NCD for the next segment. The third entry on the line (NCD) is the number of points in the drag table. If SEP_TABL is false, set NCD = NCL unless NCL is zero (indicating reuse of a previous table). If NCL is zero, set NCD equal to the segment number for the table you wish to reuse.

The next line is for the stall angle of attack. It is used for the Viterna method of producing post-stall aerodynamic data. I prefer to provide lift and drag tables that go well beyond the post-stall region so that I know exactly what aerodynamic data WT_Perf is using in the calculations. If you don't have post-stall data, you can use the Viterna method to fill in the missing data.

I_SHFT is an index that indicates the line of the C_L curve where the C_L curve departs from linear. It is used only for the 3-D delayed-stall algorithms, which should not be used. I usually set I_SHFT to 1. What follows next is one or two tables, depending on the value of the SEP_TABL flag. If SEP_TABL is true, enter NCL lines of alpha and lift coefficient followed by NCD lines of alpha and drag coefficient. If SEP_TABL is false, enter NCL lines of alpha, C_L and C_D .

If SEG is less than N_SEG, enter another SEG, NCL, NCD line, followed by the appropriate information as described above. If SEG is equal to N_SEG, the input file is complete.

CONVERTING PROP-PC FILES

To help you convert your old input files from PROP-PC format to WT_Perf format, I've created a support program called PROP2WTP. The syntax is:

```
PROP2WTP <PROP file> <WT_Perf file>
```

You'll need to go into the resulting file and add a few values that WT_Perf needs that PROP-PC did not. Look for the string "(set me)" to see what needs setting. The new variables are the thickness (or thickness to chord ratio), Alpha stall, and the index into the C_L table

where the linear portion of the life curve begins to break.

POSSIBLE FUTURE CHANGES

- I'd like to convert the code to free-form Fortran.
- I'd like to use run-time memory allocation so that we don't have to recompile to run more iterations.
- I'd like to use the AeroDyn algorithm for the induction calculations.
- I'd like to clean out Montgomerie's 3-D delayed-stall algorithm and replace it with the Corrigan and UIUC algorithms.

KNOWN BUGS

- The induction algorithm really isn't valid of skewed flow, which happens when a non-zero shaft tilt is used. The algorithm from AeroDyn should be used instead.

Caveats

NREL makes no promises about the usability or accuracy of WT_Perf, which is essentially a beta code. NREL does not have the resources to provide full support for this program. *You may use WT_Perf for evaluation purposes only.*

ACKNOWLEDGEMENTS

WT_Perf was written by Marshall Buhl of the National Wind Technology Center (NWTC). Funding for WT_Perf came from the US Department of Energy under contract to the National Renewable Energy Laboratory.

FEEDBACK

If you have problems with WT_Perf, please contact Marshall Buhl. If he has time to respond to your needs, he will do so, but please do not expect an immediate response. Please send your comments or bug reports to:

Marshall L. Buhl, Jr.
NWTC/3811
National Renewable Energy Laboratory
1617 Cole Blvd
Golden, CO 80401-3393
United States of America
<http://wind2.nrel.gov/designcodes/>
mailto:marshall_buhl@nrel.gov
Voice: (303)384-6914
Fax: (303)384-6901